# PATENT SPECIFICATION (11) 1 576 276

(19)

## (54) IMPROVEMENTS IN OR RELATING TO A DATA PROCESSING SYSTEM COMPRISING A PLURALITY OF PROCESSORS

(71) We, SIEMENS AKTIENGESELL-SCHAFT, a German Company, of Berlin and Munich, Federal Republic of Germany, do hereby declare the invention,
5 for which we pray that a patent may be granted to us, and the method by which it is to be performed to be particularly described in and by the following state-ment—:

10 This invention relates to a data process-ing system comprising a plurality of processors which differ from one another in respect of the availability of hardware devices for particular functions, and in
15 which when a processor encounters a command requiring facilities the processor does not possess, the command can be transferred to another processor for execution.

20 Arrangements which operate in this way are known for example from U.K. Patents No. 1,327,779, 1,150,110 and 1,183,158. These known arrangements however need the assistance of a superordinate operating
25 programme or a superordinate control system in order to delegate commands not executable by one processor to another processor which is capable of executing them. In the case of the assistance of a
30 superordinate operating programme one has the disadvantage that the inequality of the individual processors must be taken into account at the machine-language pro-gramming level. This means, for different
35 system configurations, an adaptation of the operating system to the processors being used. On the other hand, superordinate control systems, which operate with or without the assistance of the operating
40 programme, require an additional outlay.

A frequently occurring example of pro-cessors having different function units are those with integrated input-output control units. Whereas independent input-output
45 control units can generally be reached direct from any processor, an integrated input-output control unit can only be approached via the processor which it is assigned.

According to the presnet invention there 50 is provided a data processing system in-cluding a pair of processors which differ from one another in respect of the avail-ability of hardware devices for particular functions, wherein the sytem is arranged, 55 in operation, so that when a processor of the pair recognises a command requiring such a device not available therein but available in the other processor of the pair, it passes to said other processor an 60 instruction and all the necessary parameters for execution of the required function, and said other processor interrupts the process-ing of its own programme at the next interrupt point, executes the instruction in 65 accordance with the command parameters and returns the results to the first pro-cessor, communication of said instruction, parameters and results taking place directly without the participation of any super- 70 ordinate control system, whereby the differences between the processors of the pair is not apparent at machine language level even in the execution of commands requiring such hardware devices. 75

In a preferred embodiment the system is arranged in operation so that in the event of an overlapping occurrence in both pro-cessors of a command requiring the facilities of the other processor the 80 command counter of one processor is retarded so as to abandon execution of the command occurring therein, and said one processor executes the command trans-mitted by the other processor and repeats 85 the transmission of said abandoned command to the other processor.

The invention will be further understood from the following description by way of example of an embodiment thereof, in 90

which the processors have integrated input-output control units, with reference to the accompanying drawing, in which:—

Figure 1 schematically illustrates a simplified block diagram of a multi-processor system; and

Figures 2, 3a, and 3b are time diagrams of command sequences.

Figure 1 illustrates a multi-processor system having two processors CPU1 and CPU2, each of which is permanently assigned a respective one of input-output control units IOC1 and IOC2. It is assumed that each input-output control unit is integrated into the processor which it is assigned, this being illustrated in Figure 1 by a broken boundary line between each processor and its assigned input-output control unit. Each input-output control unit is assumed to be able to supply four input-output channels IO1 to IO4 and IO5 to IO8. The processors CPU1 and CPU2 have common access to a working or main store MM, and exchange information with one another via a multiple link IF.

If for example in a command sequence being processed by the processor CPU1 there occurs a command which the processor CPU1 interprets as being able to be handled only via for example the input-output channel IO8, then the processor CPU1 emits to the processor CPU2 an instruction (request signal) together with all the command parameters which the latter requires to execute the command. The processor CPU2 generally receives this instruction at the next interruptable point of its own command sequence, executes the command with the parameters which it has been given until the desired results are achieved, and reports these results to the processor CPU1. It does not use the results itself, and in particular does not store them. The processor CPU1, which has been in a waiting state from the emission of the instruction until the report-back of the results, employs the fed-back results as if they were its own results, stores them for example, and completes the execution of the command.

The described sequences are schematically illustrated in Figure 2, in which vertical lines under headings CPU1 and CPU2 represent commands or programme components in machine language which are executed by the processors. The interruptable points in the command sequences are marked by short transverse lines. It has been assumed that after a time TJ, the processor CPU1 recognises that the complete execution of a command B1$n$ requires a particular function unit which is integrated in the processor CPU2. Therefore the processor CPU1 transmits an instruction to the other processor CPU2 which, following the next interruptable point in its command sequence, is received and executed by the latter. The processor CPU2 supplies the results of the command execution to the processor CPU1 which then completes the command execution. Both processors can then continue with the processing of their own command sequences.

If, as is assumed here, two processors each possess a particular function unit, the situation can occur that each processor approaches the other processor with an instruction within the length of time required for the execution of corresponding commands. If no special provisions were made, in this situation both processors would wait for their instructions to be processed by the other processor. This would then lead to a mutual blocking of the two processor.

Two such conflict situations are represented in Figures 3a and 3b. In accordance with Figure 3a, after a time T1 which the processor CPU1 requires to interpret a command, this processor recognises that it is necessary to use a particular function unit in the processor CPU2 to execute the command. However, the relevant instruction to the processor CPU2 does not arrive there until an analogous command is already in progress in this processor, which command itself requires a particular function unit in the processor CPU1. Therefore the processor CPU2 directs an instruction to the processor CPU1 after the expiration of its own interpretation time T2.

In order to avoid a mutual blocking of the processors in such a conflict situation the following measures are adopted:—

a) For example by appropriate micro-programming, it is ensured that during the periods of time T1 and T2 which the processors require for the interpretation of commands, no irreversible changes occur in data, i.e. no data are modified either in the store MM or in registers; and

b) One of the two processors, for example the processor CPU1, having emitted an instruction, waits for at least the period of time T2 which the other processor CPU2 requires to interpret a command. If during this period of time the one processor CPU1 receives no instruction from the other processor CPU2 or if it receives back the results of its own instruction, the one processor CPU1 realises that the other processor CPU2 has not submitted an instruction. However, if an instruction arrives from the other processor CPU2 during the additional waiting time T2 of the one processor CPU1, the one processor CPU1 decrements its command counter. The one processor CPU1 thus

arrives at an interruptable point in its command sequence and can process the instruction from the other processor CPU2. When this instruction has been executed, it
5 again emits its own instruction to the other processor CPU2. Generally, having reached an interruptable point in its command sequence, this processor CPU2 is now in a position to execute this command. Other-
10 wise the above-described process is repeated.

Another example of the occurrence of a conflict situation is shown in Figure 3b, in which shortly after an interruptable point
15 in the command sequence of the processor CPU1, the processor CPU2 recognises a command which triggers an instruction to the processor CPU1. The latter recognises the alien instruction immediately following
20 the emission of its own instruction. Therefore it can, as before, immediately turn back its command counter, and thus return to the previous interruptable point in its command sequence without an additional
25 waiting time.

It should be expressly mentioned that the turning back of the command counter is only allowed to occur in one processor in a conflict situation.
30 WHAT WE CLAIM IS:—

1. A data processing system including a pair of processors which differ from one another in respect of the availability of hardware devices for particular functions,
35 wherein the system is arranged, in operation, so that when a processor of the pair recognises a command requiring such a device not available therein but available in the other processor of the pair, it passes to
40 said other processor an instruction and all the necessary parameters for execution of the required function, and said other processor interrupts the processing of its own programme at the next interrupt point, ex-
45 ecutes the instruction in accordance with the command parameters and returns the results to the first processor, communication of said instruction, parameters and results taking place directly without the
50 participation of any superordinate operating programme or superordinate control system, whereby the differences between the processors of the pair is not apparent at machine language level even in the ex-
55 ecution of commands requiring such hardware devices.

2. A data processing system according to claim 1, arranged in operation so that in the event of an overlapping occurrence in
60 both processors of the pair of a command requiring the facilities of the other processor the command counter of one processor is retarded so as to abandon execution of the command occurring therein,
65 and said one processor executes the command transmitted by the other processor and repeats the transmission of said abandoned command to the other processor.

3. A data processing system substan-
70 tially as herein described with reference to the accompanying drawing.

G. F. REDFERN & CO.,
Chartered Patent Agents,                75
Marlborough Lodge,
14 Farncombe Road,
Worthing BN11 2BT.

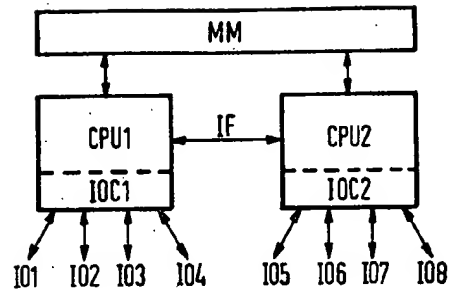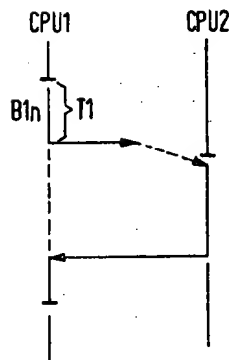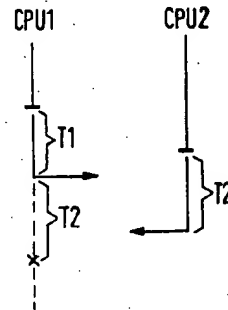Agents for the Applicants              80

Fig. 1



Fig.2



Fig.3a



Fig. 3b